

# Meritence and Polymorphism

## Inheritance

- CLASS Mom  
attr\_reader :name, :work

def initialize(name, work)

@name = name

@work = work

end

end

CLASS Chicken

attr\_reader :screw\_up

def initialize(screw\_up)

@screw\_up = screw\_up

super(name)

end

end

CLASS Mother



```
class Animal
  attr_reader :name

  def initialize(name)
    @name = name
  end

  def communicate
    print "Hi, I'm the SUPER class, or the mom class!!"
  end
end

class Dog < Animal
  attr_reader :race

  def initialize(name, race)
    super(name)
    @race = race
  end
end

shitzo = Dog.new("Caramelo", "Shitzo")

puts "Dog name: #{shitzo.name}, dog race: #{shitzo.race}"
puts shitzo.communicate
```

Super  
CLASS

Chicken  
CLASS

Polymorphism  
The class  
Dog extend the  
function  
communicate

## Polymorphism Whitting

```
class Dog < Animal
  attr_reader :race

  def initialize(name, race)
    super(name)
    @race = race
  end

  def communicate
    puts "au, auau, wtf, auau"
  end
end

shitzo = Dog.new("Caramelo", "Shitzo")

puts "Dog name: #{shitzo.name}, dog race: #{shitzo.race}"
puts shitzo.communicate
```

Same  
Name

## Overload

```
class Calculator
  def somar(*args)
    arr_integer = []
    arr_integer.push(*args)
    arr_integer.inject(:+)
  end
end

c1 = Calculator.new

puts c1.somar(2, 5, 5, 5)
```

ADD ALL ARGS  
To list  
Sum all items

return 17